Reproducible Research: Assesment 1

Diego Santana

20/8/2020

knit options setup

Introduction

This is an R Markdown document, developed for the Coursera course "Reproducible Research".Based on the following characteristics of the data.

Now is possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. These type of devices are part of the "quantified self" movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data.

This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

The data for this assignment can be downloaded from the course web site: https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip
Activity monitoring data [52K]

- The variables included in this dataset are:
- steps: Number of steps taking in a 5-minute interval (missing values are coded as NA)
- date: The date on which the measurement was taken in YYYY-MM-DD format
- interval: Identifier for the 5-minute interval in which measurement was taken

Loading and preprocessing the data

Set the working directory, downloading and unzipping the file to "step_data.csv".

```
knitr::opts_chunk$set(warning=FALSE)
library(ggplot2)

setwd("C:/Users/ALs/Desktop/Coursera/CourseraCienciaDatosR/05_Reproducible Research/Assesment1")
url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
destfile <- "step_data.zip"
download.file(url, destfile)

unzip(destfile)
```

```
activity <- read.csv("activity.csv", sep = ",")
```

The the structure and variable names of the file are:

```
names(activity)
```

```
## [1] "steps"    "date"     "interval"
```

```
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

```
head(activity[which(!is.na(activity$steps)), ]) # data set with NA rows removed
```

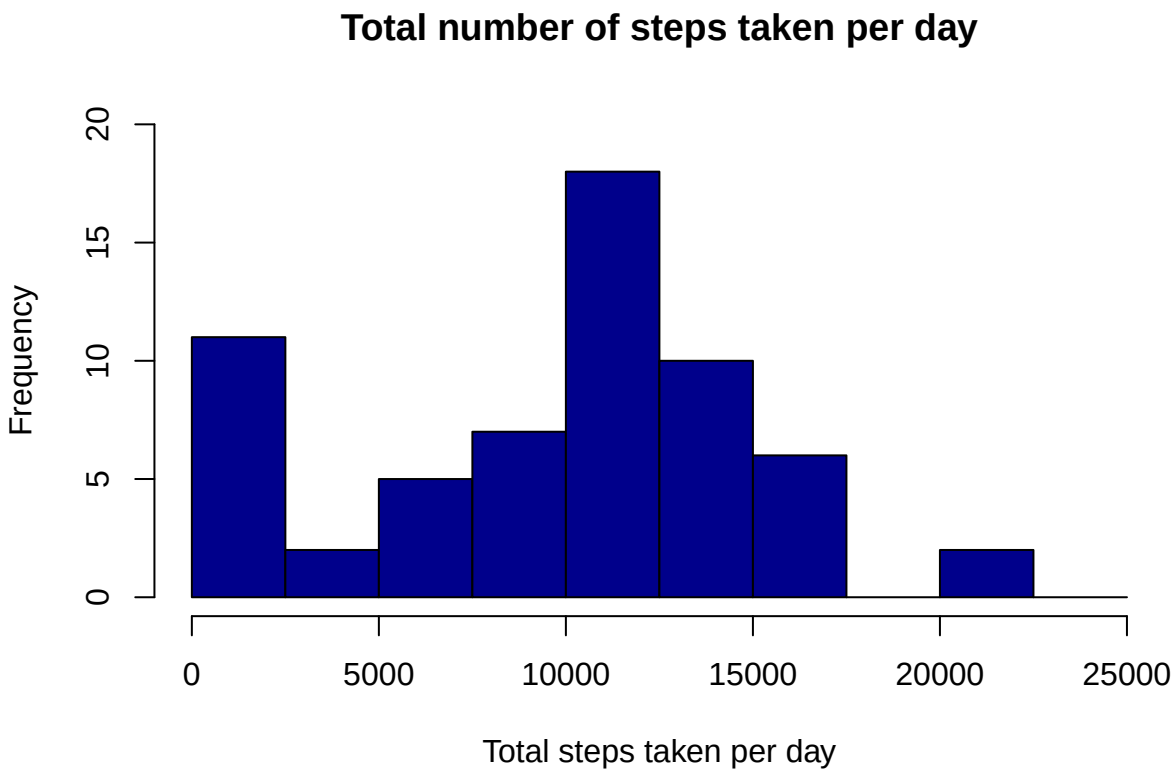
```
##     steps       date interval
## 289     0 2012-10-02        0
## 290     0 2012-10-02        5
## 291     0 2012-10-02       10
## 292     0 2012-10-02       15
## 293     0 2012-10-02       20
## 294     0 2012-10-02       25
```

Now the file is ready for analysis.

What is mean total number of steps taken per day?

- Group the number of steps by date and intervals.
- Find the total number of steps per day over all days.
- Remove N/A rows for this part. #### Histogram of the total number of steps taken each day.

```
activity_total_steps <- with(activity, aggregate(steps, by = list(date), FUN = sum, na.rm = TRUE))
names(activity_total_steps) <- c("date", "steps")
hist(activity_total_steps$steps, main = "Total number of steps taken per day", xlab = "Total steps taken
```




```
summary(activity_total_steps$steps)
```

The summary of 'total number of steps per day'.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    6778   10395    9354   12811   21194
```

```
mean(activity_total_steps$steps)
```

The maen of 'total number of steps per day'.

```
## [1] 9354.23
```

```
median(activity_total_steps$steps)
```

The median of 'total number of steps per day'.

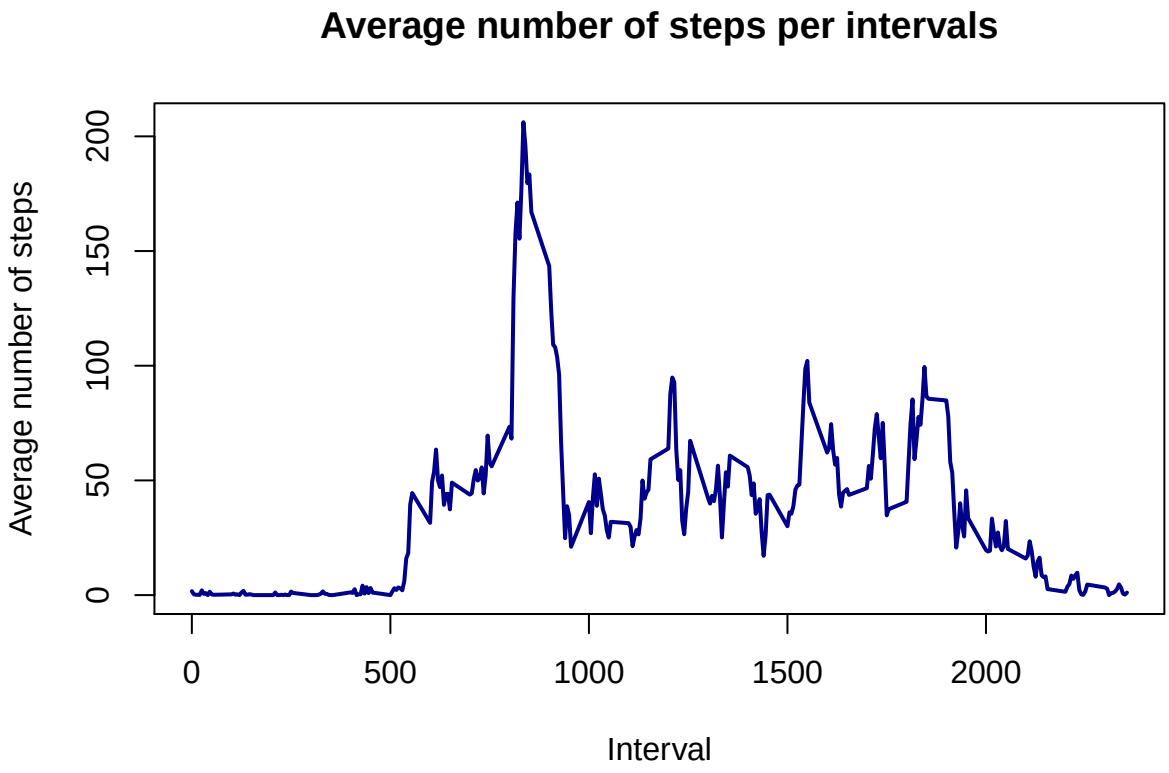
```
## [1] 10395
```

What is the average daily activity pattern?

In this section, we make a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken averaged across all days.

```
average_daily_activity <- aggregate(activity$steps, by=list(activity$interval), FUN=mean, na.rm=TRUE)
names(average_daily_activity) <- c("interval", "mean")

plot(average_daily_activity$interval, average_daily_activity$mean, type = "l", col="darkblue", lwd = 2,
```



The time interval during which the maximum number of steps is taken is

```
average_daily_activity[which.max(average_daily_activity$mean), ]$interval
```

```
## [1] 835
```

Imputing missing values

First of all, let us get a sense for the missing values. Are there days with all time intervals reporting NA step values?

```
sum(is.na(activity$steps))
```

Calculate the number of rows with NA values

```
## [1] 2304
```

```
sum(is.na(activity$steps))*100/nrow(activity)
```

Percentage of rows with missing values

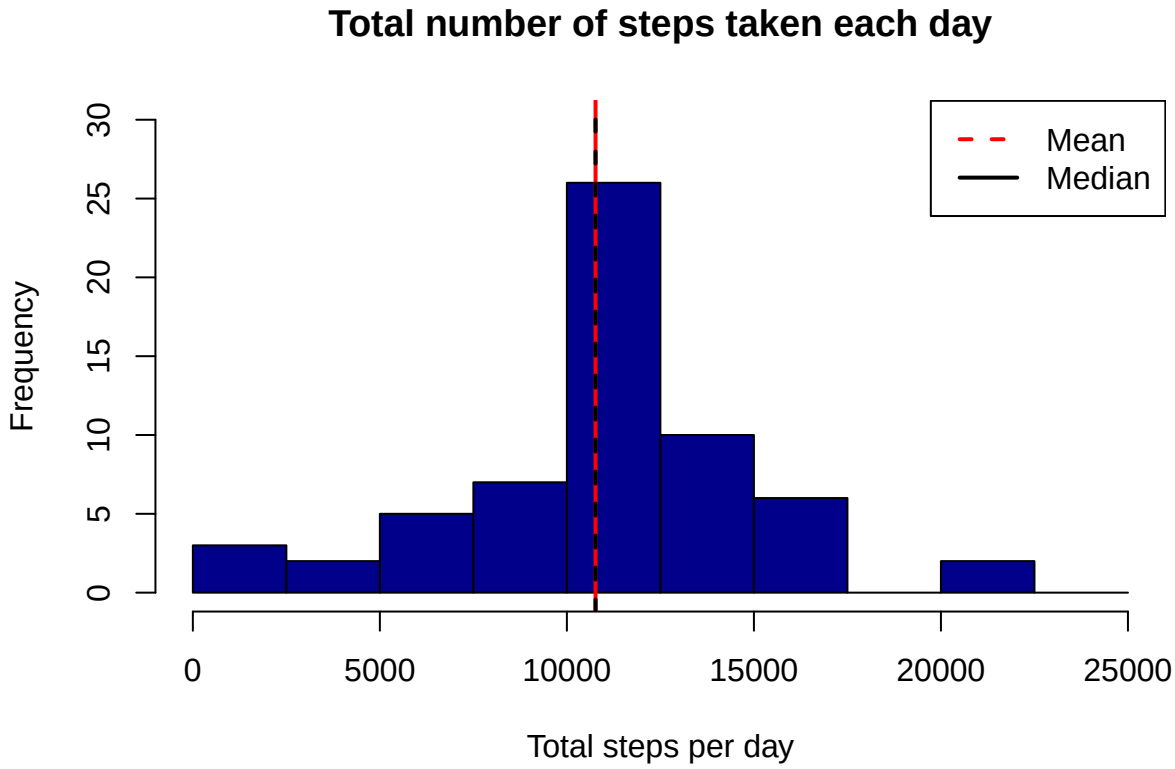
```
## [1] 13.11475
```

Replace the missing data for a day by the time average over all other days. Create a new dataset that is equal to the original dataset but with the missing data

```
imputed_steps <- average_daily_activity$mean[match(activity$interval, average_daily_activity$interval)]
activity_imputed <- transform(activity, steps = ifelse(is.na(activity$steps), yes = imputed_steps, no =
total_steps_imputed <- aggregate(steps ~ date, activity_imputed, sum)
names(total_steps_imputed) <- c("date", "daily_steps")
```

```
hist(total_steps_imputed$daily_steps, col = "darkblue", xlab = "Total steps per day", ylim = c(0,30), ma
abline(v = mean(total_steps_imputed$daily_steps), lty = 1, lwd = 2, col = "red")
abline(v = median(total_steps_imputed$daily_steps), lty = 2, lwd = 2, col = "black")
legend(x = "topright", c("Mean", "Median"), col = c("red", "black"), lty = c(2, 1), lwd = c(2, 2))
```

Histogram of the total number of steps taken each day with the imputed missing values.




```
mean(total_steps_imputed$daily_steps)
```

The maen of 'total number of steps per day'.

```
## [1] 10766.19
```

```
median(total_steps_imputed$daily_steps)
```

The median of 'total number of steps per day'.

```
## [1] 10766.19
```

Are there differences in activity patterns between weekdays and weekends?

Create a new column describing if the date is a weekday or weekend.

```
activity$date <- as.Date(strptime(activity$date, format="%Y-%m-%d"))
activity$datetype <- sapply(activity$date, function(x) {
        if (weekdays(x) == "Saturday" | weekdays(x) =="Sunday")
                {y <- "Weekend"} else
                {y <- "Weekday"}
                y
        })
```

Steps taken over each interval averaged across weekday days and weekend days.

```
activity_by_date <- aggregate(steps~interval + datetype, activity, mean, na.rm = TRUE)
ggplot(activity_by_date, aes(x = interval , y = steps, color = datetype)) +
       geom_line() + facet_wrap(~datetype, ncol = 1, nrow=2) +
       labs(title = "Average daily steps by type of date", x = "Interval", y = "Average number of steps
```